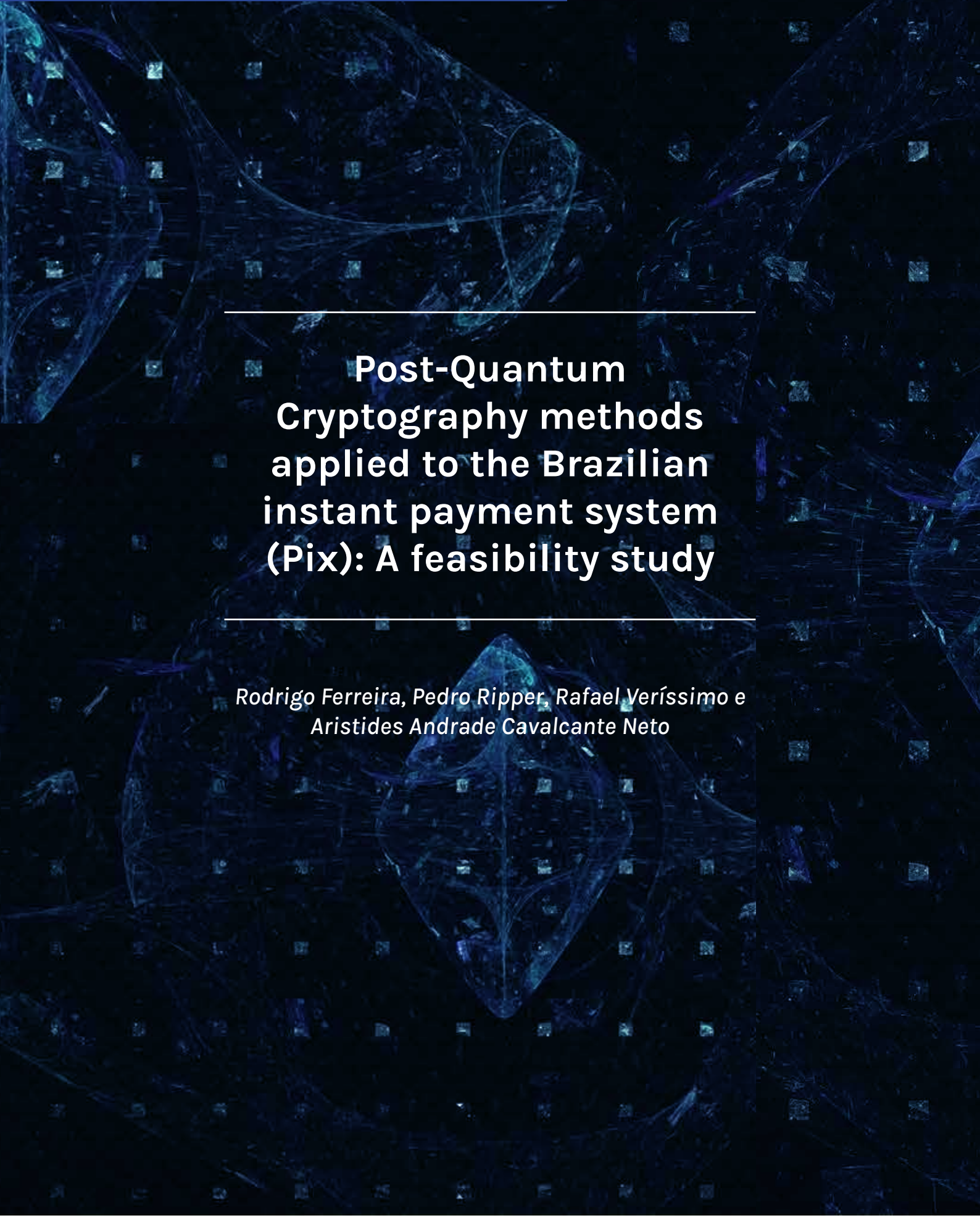


Post-Quantum Cryptography methods applied to the Brazilian instant payment system (Pix): A feasibility study

Rodrigo Ferreira, Pedro Ripper, Rafael Veríssimo e Aristides Andrade Cavalcante Neto

Authors

BANCO CENTRAL DO BRASIL

Brazil Quantum

Partner

Microsoft

Supporter

Fenasbac

Post-Quantum Cryptography methods applied to the Brazilian instant payment system (Pix): A feasibility study

Rodrigo Ferreira[1], Pedro Ripper[1], Rafael Veríssimo[1] e Aristides Andrade Cavalcante Neto[2]

E-mail
contato@brazilquantum.com

Research coordinator
Aristides Andrade Cavalcante Neto

1 *Brazil Quantum, São Paulo, SP, Brazil*
2 *Banco Central do Brasil, Brasília, DF, Brazil*

Post-Quantum Cryptography methods applied to the Brazilian instant payment system (Pix): A feasibility study

Aristides Andrade Cavalcante Neto (aristides.andrade@bcb.gov.br)
Pedro Ripper (pedro.ripper@brazilquantum.com)
Rafael Veríssimo (rafael.verissimo@brazilquantum.com)
Rodrigo Ferreira (rodrigo.ferreira@brazilquantum.com)

Abstract

The development of Quantum Computing in the last few years has not only created new opportunities, but also raised concerns about the current state of cryptography. Followed by the threat that quantum algorithms pose to some of the most used cryptographic systems, a new field of research has been created. Post-Quantum Cryptography (PQC) is responsible for the study of cryptographic methods resistant to potential "quantum attacks". This work analyses the feasibility of implementing PQC algorithms in the Brazilian Instant Payment System (Pix). To achieve such goal, we first review the present state of PQC and Pix's cryptography. Afterwards, we execute performance tests for the PQC Picnic scheme, followed by their results and the discussion on future steps for the implementation of PQC systems on Pix.

1. Introduction

In the last few years, the investment in Quantum Computing (QC) has presented a surge in its growth. The expectations on what a quantum computer is capable of have been attracting new investors and causing the creation of promising new companies. For instance, the startup IonQ which, in 2021, became the first QC publicly listed company with a 2 billion dollars evaluation [16].

The current stage of QC is characterized by machines that are very limited by noisy operations and interactions with the environment. This phase is known as the Noise Intermediate-Scale Quantum (NISQ) Era [17]. Having said that, quantum computers are yet to present any real-world application. However, the research on QC has been advancing quickly and presenting solutions for these shortcomings, turning some use cases for QC feasible in the next 5 to 10 years [22].

This rapid growth is also linked to the threat that quantum computers may pose, in the future, to current cryptographic systems. It has been shown, theoretically, that some quantum algorithms are capable of cracking certain protocols. For example, the Shor's Algorithm [18] for prime number factorization is able to solve the widely used RSA cryptosystem, with a sufficient number of fault-tolerant qubits.

As previously stated, in the NISQ Era [17], quantum computers still do not represent a risk for cryptography. However, given that there is a possibility of security breaches, researchers are developing cryptographic schemes that are resistant to quantum algorithms.

In this regard, Post-Quantum Cryptography (PQC), also known as quantum safe or quantum resistant encryption, presents itself as a solution to this issue. This new research area studies cryptosystems based on mathematical problems that are unsolvable even for current quantum algorithms, unraveling quantum-resistant cryptographic algorithms.

Some already known cryptographic systems, alongside new ones that are being developed, constitute the group of post-quantum algorithms. Taking that into account, the National Institute of Standards and Technology (NIST), created a Standardization Process for PQC, in partnership with enterprises and universities around the globe [6].

Following this international effort and the technology innovation essence of the Central Bank of Brazil (CBB), it was presented the idea of a study for the feasibility of applying some of the PQC algorithms analyzed by NIST into the Pix system (Brazilian instant payment system), developed by CBB. This work was done by Brazil Quantum, with the support of Microsoft Brazil and the technology team of CBB.

The study was initiated with the selection of NIST's PQC algorithms that were aligned with Pix's ground rules, with the following criteria: security, performance and cryptographic-agility. Afterwards, it was carried out some test cases, taking as benchmark the standard message traffic from Pix. Therefore, it was possible to make a comparison between the current Pix's performance and the one that would be obtained in a scenario with PQC algorithms.

This article begins with a brief introduction to PQC, followed by a review of its current state. Subsequently, Pix's cryptographic system is presented, as well as the performance assessment. Finally, a discussion upon the results from the PQC protocols in contrast with Pix's algorithms is presented. As conclusion, a roadmap for the incorporation of PQC algorithms in Pix is shown.

2. Fundamentals of Post-Quantum Cryptography

As previously presented, PQC is focused on algorithms that are believed to be quantum safe. That is to say, they are currently considered resistant to quantum computers. With that in mind, it is still unknown the true capability of a quantum computer. The advancements on the research of novel quantum algorithms could make some of the PQC algorithms vulnerable in the future.

Such scenario highlights the relevance of the work being done by NIST [6], in which several PQC algorithms are being analyzed since the end of 2016. These algorithms can be categorized according to two aspects. The first one being their purpose, which can be Public Key Encryption or Digital Signature. Moreover, they can be divided into the mathematical problem that they are based on. Therefore, some of the groups of PQC algorithms are:

- **Hash-Based:** cryptographic systems with their security based on the collision-resistance and the inversibility of their hash functions. Hash-Based cryptography is widely used in Digital Signature schemes. Relying basically on a secure hash function, Hash-Based Digital Signature systems do not present relevant computational cost. A good example of a Hash-Based algorithm is SPHINCS+ [19,20,21].
- **Code-Based:** this group of algorithms is based on the theory of Error Correcting codes. According to Enisa's report [20], Code-Based protocols were developed to offer short signatures at the cost of generating larger key sizes. Some of the algorithms that belong to this group are Classic McEliece, BIKE e HQC [19,20].
- **Lattice-Based:** this class of algorithms makes use of mathematically hard problems inside the study of lattices. Lattice-Based algorithms, such as NTRU, Saber and Frodo-KEM [19,20], account for the majority of PQC groups in NIST'S Third Round of standardization of PQC.

Furthermore, the PQC algorithms existent today are not sufficient to guarantee quantum resistance. Besides the aforementioned fact that more powerful quantum algorithms may be developed in the future, most PQC algorithms are not ready to be implemented in the real world. These protocols still have some points for improvement, such as their efficiency, security, and portability regarding different technologies.

Therefore, NIST's Standardization Process [6] is of great importance, playing a part in the development of PQC and selecting the most fitting algorithms to prevent threats that quantum algorithms may pose.

3. Current state of Post-Quantum Cryptography

The current progress of PQC is greatly influenced by NIST's Standardization Process [6]. This project is based on rounds in which different algorithms are submitted to be evaluated in order to find shortcomings and elements that can be refined. The proposed cryptographic systems are open-source projects with their documentation available to anyone, in order to democratize the access for testing and research.

Having its first round in the end of 2016, NIST's Standardization Process is found today in its third stage. During this period, some algorithms were discarded, due to weaknesses that were found, while others prevailed and continued to be analyzed and improved. However, the winning candidates in each round can still be rejected in future phases, on account of new vulnerabilities that can be found.

4. Pix's cryptographic system overview

The main goal of the Brazilian instant payment system (Pix) is to maintain a secure system for its operation. In order to guarantee such security, it is necessary to stablish a set of protocols that define the interactions between all the segments of the payment system [2].

Aspects of these protocols contemplate communication cryptography, authentication, digital signature processes and managing digital certificates. Moreover, audit logs must also be stored in order to provide traceability of transactions inside the Pix network [1].

With that said, the communication between each Payment Service Provider (PSP) and Pix's APIs is performed through the RSFN (Brazilian Financial System Network). That communication must follow the rules stated in the Networks' Manual of the National Financial System [2].

The connection between a PSP and the APIs available on Pix occurs via the Hypertext Transfer Protocol (HTTP) 1.1, using the Transport Layer Security (TLS) cryptography version 1.2 or above, with a mandatory mutual authentication at the moment of connection. The algorithms contemplated in this cryptography are presented in Table 1.

Table 1. Functions and their respective algorithms in the TLS cryptography used by Pix.

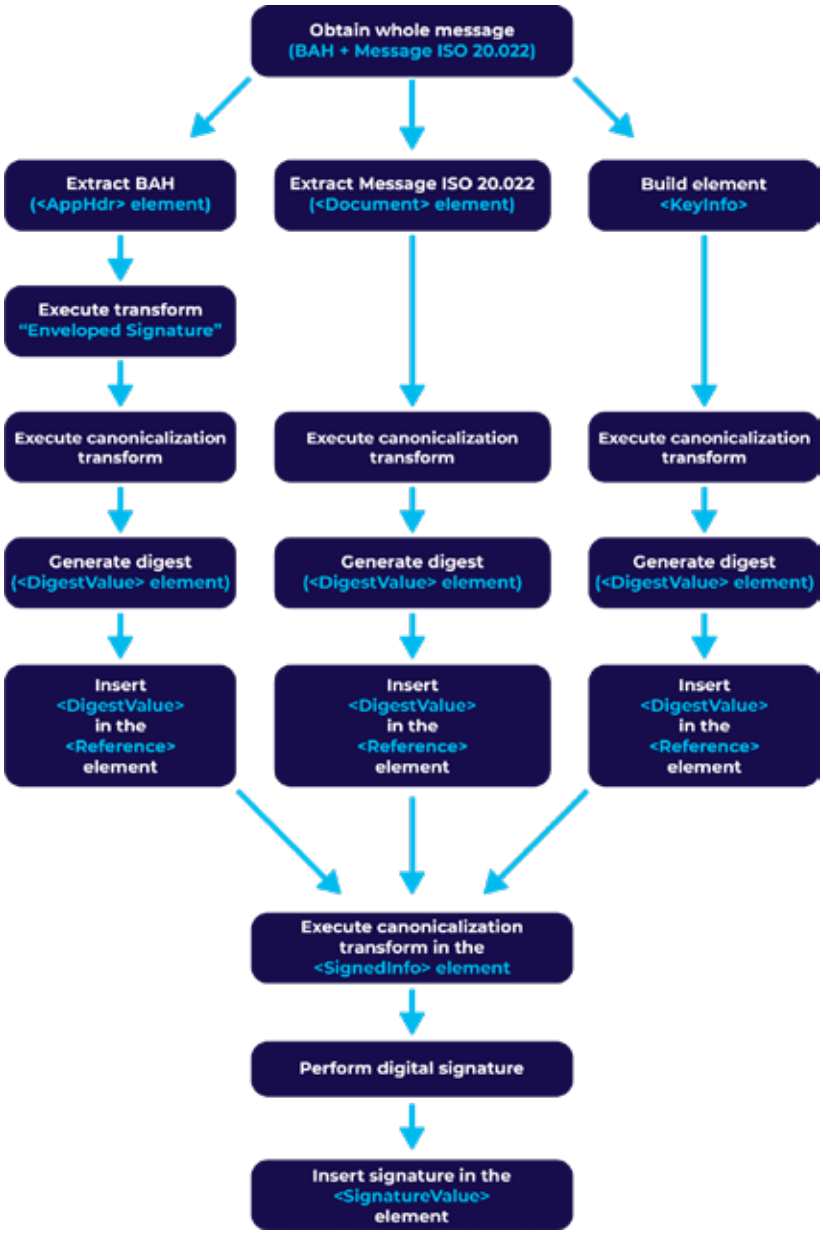
Function	Algorithm
Key Exchange	ECDHE
Authentication	RSA
Symmetric Cryptography	AES-128, modo GCM
MAC (*Message Authentication Code*)	SHA-256

It's necessary to state that CBB and PSP must use ICP-Brazil certificates. Moreover, the PSP's HTTP clients need to satisfy the TTL (Time to Live) definitions of the DNS servers, in order to assure the access to Pix's APIs at all times.

To safeguard Pix's transactions, the transmitter digitally signs all messages sent to SPI (Portuguese acronym for "Instant Payment System" – the core module of Pix) [1]. For every operation (considering the different message types [1,2]), the CBB's answer to the PSP is always digitally signed.

In the context of Pix, the digital signature standard is XMLDSig [4]. In addition, in SPI the messages follow the ISO 20.022 standard [5]. In this sense, the information that must be signed are: the ISO 20.022 message itself (<*Document*>), the BAH header (<*AppHdr*>), and the <*KeyInfo*>. Figure 1 illustrates the digital signature process of IPS messages.

Figure 1. SPI's digital signature workflow for messages.



5. Post-Quantum Cryptography applied to Pix

The feasibility study about the application of PQC into Pix was conducted using NIST's PQC Standardization Process [6] as reference. In its second round, NIST selected 26 algorithms in total. As key exchange protocols, the chosen ones were Classic McEliece, CRYSTALS-KYBER, NTRU, SABER, BIKE, FrodoKEM, HQC, NTRU Prime, SIKE, LAC, LEDAcrypt, NewHope, NTS-KEM, ROLLO, Round5, RQC and Three Bears. Moreover, the following digital signature algorithms were elected: CRYSTALS-DILITHIUM, FALCON, Rainbow, GeMSS, Picnic, SPHINCS+, LUOV, MQDSS and qTESLA [6].

In the third round of the NIST standardization process, 15 algorithms were selected and categorized as "finalist" or "alternative" candidates. The "finalists" are algorithms that NIST considers to have the greatest potential to become the standard by the end of the

third round. The " alternative " candidates, in turn, are perceived by NIST as potential future standards after further rounds of evaluation [14]. The list of finalist and alternative candidates can be seen in tables 2 and 3, respectively [14].

Table 2. Round 3 finalists.

Key Exchange	Digital Signature
Classic McEliece	CRYSTALS-DILITHIUM
CRYSTALS-KYBER	FALCON
NTRU	Rainbow
SABER	

Table 3. Round 3 alternate candidates.

Key Exchange	Digital Signature
BIKE	GeMSS
FrodoKEM	Picnic
HQC	SPHINCS+
NTRU Prime	
SIKE	

Aligned with the evaluation criteria highlighted by NIST, the authors considered the following parameters in their study:

- **Safety**: the most important factor for NIST [6], which has defined five levels of safety based on the amount of computing resources needed to perform a brute force attack;
- **Cost and performance**: the second most relevant aspect, considering computational and data transfer costs, as well as the performance of algorithms in key generation or digital signature authentication;
- **Crypto-agility**: crucial factor for the implementation of post-quantum cryptography in Pix, since the selected algorithms must be easily implemented into existing systems (the transition must be facilitated).

Given such conditions and the support offered by Microsoft, Brazil Quantum chose to analyze the algorithms developed by *Microsoft Research* team: FrodoKEM, SIKE, qTESLA (key exchange) and Picnic (digital signature) [7].

After joint deliberation with the Pix development team, we concluded that the way the system was built makes it not feasible to analyze the computational time and cost of an isolated key exchange. This occurs because, in Pix, messages are not encrypted individually, but rather at the transport layer - since they are transmitted via HTTP.

This scenario provided, therefore, the study directed to the digital signature of messages in SPI. Thus, Brazil Quantum's team was dedicated to implement the Picnic algorithm (considering Pix's current cryptographic configurations) and to analyze the results obtained based on Pix's current performance.

6. Picnic

Picnic is a digital signature quantum-safe algorithm. It has been developed by several researchers from Aarhus University, AIT GmbH, DFINITY, Georgia Tech, Microsoft Research, Northwestern University, Princeton University, Denmark Technical University, and the University of Maryland. Picnic's safety comes from a Zero-Knowledge Proof (ZKP) system, in which the transmitter can prove to the receiver that the message has been encrypted - without revealing it.

Moreover, Picnic also uses block ciphers and hash functions, enhancing its security to the post-quantum level. We've then analyzed Picnic's cryptographic components (LowMC, hash function, derivation key function) to determine which configuration fulfills Pix's needs.

- **LowMC:** parameterized block cipher used for simulating the Multi-Party Computation (MPC) protocol. We define it by the binary matrices order n and the number of LowMC rounds r;
- **Hash function:** an algorithm maps variable-length data to fixed-length output. In Picnic's context, it can be SHAKE128 or SHAKE256 - depending on the security level L;
- **Key Derivation Function (KDF):** when creating and verifying digital signatures, it is necessary to expand a small random value (seed) from 128 to 512 bits to a larger one (around 1kB), which we can achieve via a SHAKE function. For Picnic, we use the same function (SHAKE128 or SHAKE256) for hash and key derivation.

Moreover, Picnic's ZKP implementations are based on the Fiat-Shamir (FS) or the Unruh (UR) transformations, generating two types of Picnic for each security level (L). NIST has established three security levels (L1, L3, L5) that correspond to the security offered by AES-128, AES-192, and AES-256.

Our research has also analyzed Picnic's third version (Picnic3), having minor implementation differences. The following table contains the cryptographic parameters of each Picnic type, including the S security bits (S for classical attacks and, at least, S/2 for quantum attacks):

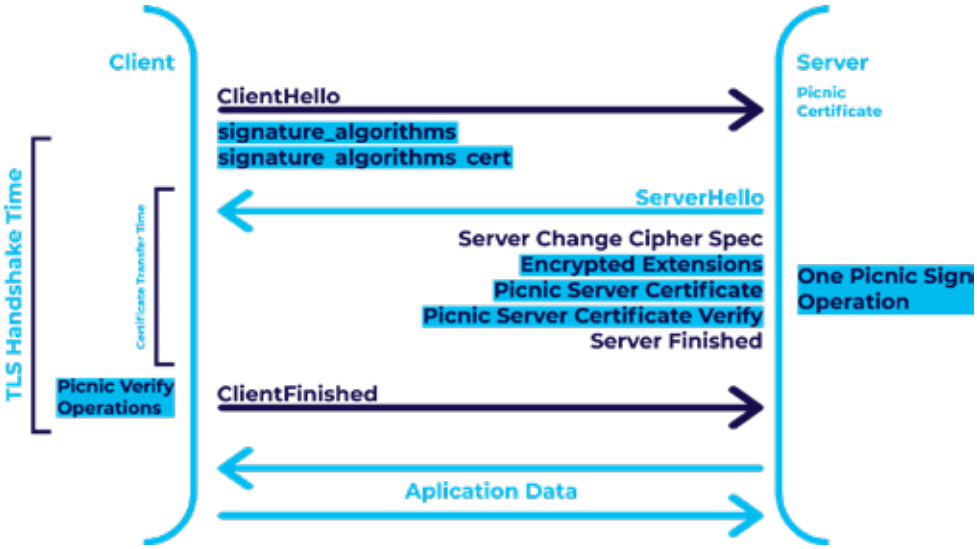
Table 4. Picnic types and cryptographic parameters.

Picnic	*S[bits]*	*n[bits]*	*r[rounds]*	*k[bits]*	*Hash/KDF*
Picnic L1	128	128	20	128	SHAKE128
Picnic L3	192	192	30	192	SHAKE256
Picnic L5	256	256	38	256	SHAKE256
Picnic3 L1	128	129	4	128	SHAKE128
Picnic3 L3	192	192	4	192	SHAKE256
Picnic3 L5	256	255	4	256	SHAKE256

The size k (Table 4) corresponds to the key size (in bits) that we use in the algorithm. Also, the cryptographic parameters remain the same for both versions of Picnic (Picnic-FS and Picnic-UR).

Figure 2 is a high-level diagram of Picnic's architecture, containing the client-server interaction. We have adopted the TLS Handshake time as the telemetry.

Figure 2. Picnic's high-level architecture, including the client-server interaction.



7. Implementation

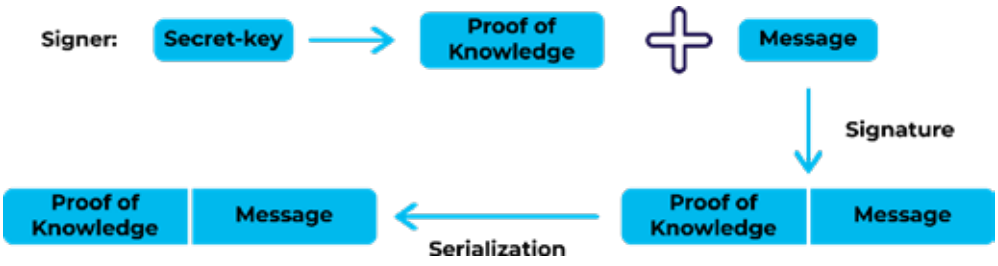
Regarding the choice between the system's possible cryptographic parameters, Picnic was implemented and tested in its Picnic L3 and Picnic3 L3 versions, since they have similarities to the current state of Pix (focused on crypto-agility). Then, some possible improvements to Pix were analyzed with the objective of making the system's digital signature process *quantum-safe*.

Such changes include changing the RSA-SHA256 algorithm for Picnic, which can be obtained by editing the <*SignatureMethod*> element in the Pix system. As indicated by [12], Picnic is expected to be a feasible option for the future of digital signatures, given it is considerably more resistant to quantum attacks than traditional methods.

The following phase was the testing of Picnic L3 (FS and UR) and PIcnic3 L3. The test was composed of four sections: key generation, message signature, signature verification and key serialization. The process is illustrated in the following Figure.

By analyzing Picnic's cryptographic parameters' options, we have implemented the Picnic L3 and Picnic3 L3 versions, given that they are similar to current Pix's configurations. Thus, we have investigated which adaptations are necessary to provide quantum-safe security to Pix's digital signature process.

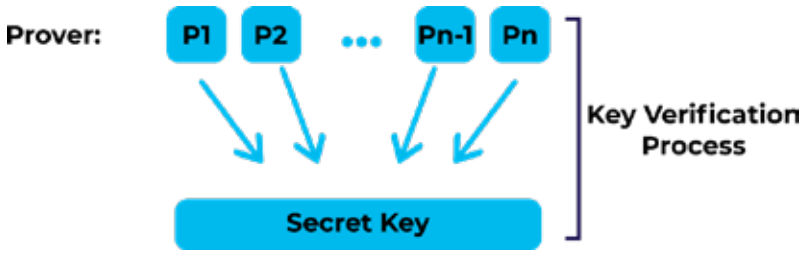
Figure 3. Picnic test procedure.



As shown in Figure 3, the *signer* (in possession of the secret key) can generate a *proof of knowledge*, which will then be combined with the message to form the digital signature. The serialization step follows, reducing the signature's occupied size in memory.

In the key verification step, a simulation in *N* parts takes place – each containing a fraction of the secret key. The fractions of each part are gathered together, forming the desired secret key. The system then checks if the secret key satisfies the *proof of knowledge* problem.

Figure 4. Key verification during Picnic tests.



In order to run the tests, it was also needed to define the messages' size. For such a task, commonly circulating messages on SPI were considered, both in case of single operation (Table 5) or multiple operations (Table 6) messages, as is the case of *Pacs. 008* (a message type widely used in this context) [1,2]. The message size difference between the minimum and maximum traffic was considered for Picnic tests.

Table 5. Common Pix messages' types and sizes.

Message Type	*Size (kB)*
Admi. 002	2,7
Pibr. 001	2,4
Pibr. 002	2,4
Pacs. 002	2,6
Pacs. 004	3,0
Pacs. 008	3,5
Camt. 040	2,7
Camt. 052	2,8
Camt. 053	3,0
Camt. 054	3,8
Camt. 060	2,7
Reda. 014	2,6

Table 6. Amount of Pacs. 008 operations and their size in KB.

Amount of operations [Pacs. 008]	Size(kB)
1	3,5
2	4,4
3	5,3
4	6,2
5	7,1

Amount of operations [Pacs. 008]	Size(kB)
6	8,0
7	8,9
8	9,8
9	10,7
10	11,6

8. Obtained Results

As shown in Tables 5 and 6, the size of commonly sent messages oscillates around 2KB and 12KB. Therefore, 100 iterations of each considered version of the system (Picnic L3 FS, Picnic L3 UR, Picnic3) were run, always with time measurement. The average time per message size was then plotted (Figures 5 and 6). It should be noted that all tested cases were successful, meaning all four steps of the test were successfully concluded.

Figure 5. Average signature time per message size.

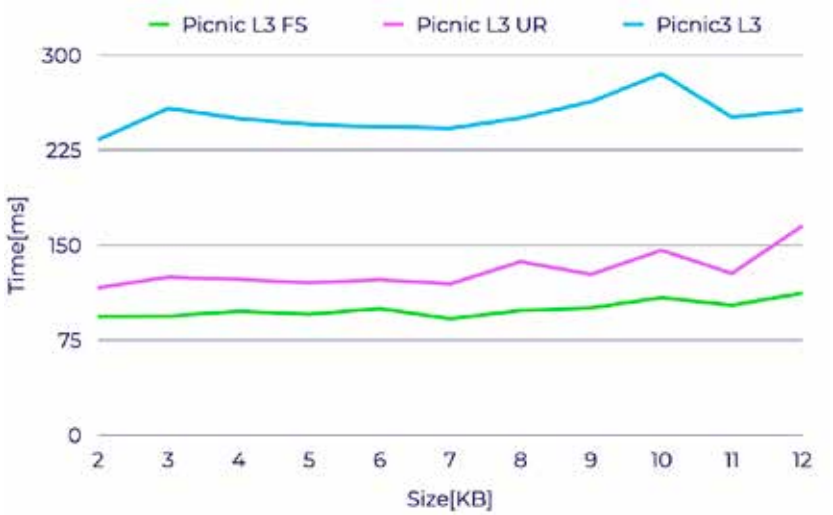
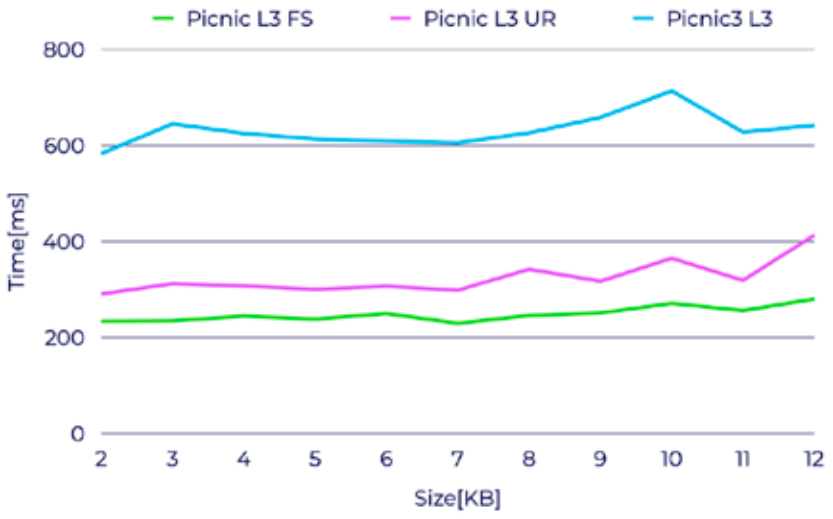


Figure 6. Average total time per message size.



The results shown on Figures 5 and 6 were obtained with a development environment with an Intel Core i7-8550U 1.99GHz processor and the GNU GCC 5.4.72 (WSL2) compiler. The user guide, as well as the testing Docker container, can be found on Brazil Quantum's page on GitHub [13].

Such results were compared with the load testing data supplied by the CBB. Such a testing data was created by the BCB during the technical specification of Pix. Therefore, the reference timing (in milliseconds) for each message type (pacs.002, pacs.004, pacs.008, admi.002, camt.052, camt.053 and pibr.002) is known, from the message reading until the writing confirmation. The measured time also includes the creation and signature of the XML message.

The test has simulated approximately 2000 tps (*transactions per second*) during 10 minutes, with different message types being sent. The minimal, average and maximal times were then obtained, as well as the P5, P50, P95 and P99 latencies. A sample of the results can be seen on Table 7.

Table 7. Properties of the messages used in the Pix environment.

Message type	Number of operations	Number of messages	Min time (ms)	Average time (ms)	Max time (ms)	P5	P50	P95	P99
Admi.002	1	524	9	23	107	13	22	31	52
Camt.052	1	9	14	21	28	15	22	26	28
Camt.053	1	342	9	26	171	15	24	45	91
Pacs.002	7	2720	11	27	171	18	25	43	60
Pacs.004	2	2	24	27	29	24	27	29	29
Pacs.008	5	1657	13	29	237	20	26	44	64
Pibr.002	1	400	12	26	199	15	24	37	70

Using the same metric (average total time), the Pix load test results stayed within a 22 to 32 milliseconds range, depending on the size of the message (ranging between 2 KB and 12 KB). In comparison to the performance of the tested Picnic versions, this approach has shown a sensibly larger total time.

9. Conclusion

The implementation of versions Picnic L3 FS, Picnic L3 UR and Picnic3 L3 versions in the Intel Core i7-8550U 1.99 GHz processor environment via the GNU GCC compiler provides results that prove incompatibility with the current demands of Pix, which requires a message throughput of at least 2000 messages per second (with an expected time of up to 50 milliseconds per message).

In this context, the message signing tests via Picnic L3 FS showed a processing time of 4 to 5 times higher than Pix's benchmark. It is also noteworthy that the results obtained in this simulation are in agreement with the updated literature on the performance of Picnic [26].

However, it is important to note that the operating conditions of the compared cryptographic systems are not the same. Internally, Pix makes use of an HSM (Hardware Security Module) from DINAMO Networks [23] that enables up to 4000 operations per second. In the current market, there are already HSM solutions supporting quantum-safe cryptography scenarios, as offered by companies like Thales [24] and Utimaco [25].

10. Discussion and Future Work

According to the results, we can see that although Picnic L3 and Picnic3 L3 provide some advantages (e.g., higher security and smaller public keys), their running times are substantially higher. In general, Picnic's approach is relatively new and is constantly evolving. It still requires further development (especially in block ciphers) before NIST can take it as a standard [14].

On the other hand, Picnic's diversity (not based on algebraic problems or complex lattices) is an advantage for future standardization. For those reasons, NIST has considered Picnic an alternate candidate [14], revealing that such an algorithm might be promising for digital signatures.

Future work could include the side-channel attacks in the Picnic scheme. Previous research [12] suggests that direct implementations would have a high concentration of side-channel attacks. We should take that into account when considering a practical application within Pix.

Moreover, to perform a consistent comparison of cryptographic schemes' performance, we could implement Picnic on Pix's hardware itself. Another possibility is replicating the same environment on Microsoft Azure [15] with the appropriate project description (objective, sizing, architecture, components, etc.). Thus, we could obtain more accurate estimates of the adoption cost of a quantum-safe API in Pix's systems.

An alternate approach is analyzing the post-quantum HSMs' compatibility with Picnic. Next, one should also consider the commercial availability and the limitations of such devices. Those applications can enhance Picnic's performance to get closer to Pix's requirements.

11. References

[1] *Manual de Segurança do Pix Versão 3.4*
URL: https://www.bcb.gov.br/estabilidadefinanceira/comunicacaodados

[2] *Manual de Redes do SFN Versão 9.2*
URL: https://www.bcb.gov.br/content/estabilidadefinanceira/cedsfn/Manual_de_Redes_do_SFN_Ver_9.2.pdf

[3] *ICP Brasil - Infraestrutura de Chaves Públicas Brasileira*
URL: https://www.gov.br/iti/pt-br

[4] *W3C Recommendation - XML Signature Syntax and Processing.*
URL: https://www.w3.org/TR/2008/REC-xmldsig-core-20080610/

[5] *ISO 20.022 Standard*
URL: https://www.iso20022.org/

[6] *NIST Post-Quantum Cryptography Standardization Process.*
URL: https://csrc.nist.gov/projects/post-quantum-cryptography

[7] *Post-quantum cryptography – Microsoft Research*
URL: https://www.microsoft.com/en-us/research/project/post-quantum-cryptography/

[8] *Picnic – Microsoft Research*
URL: https://www.microsoft.com/en-us/research/project/picnic/

[9] *Picnic – A Family of Post-Quantum Secure Digital Signature Algorithms*
URL: https://microsoft.github.io/Picnic/

[10] *The Picnic Signature Algorithm Specification*
URL: https://github.com/Microsoft/Picnic/tree/master/spec

[11] *NIST – Submission requirements and evaluation criteria*
URL: https://beta.csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf

[12] Gellersen, T., Seker, O., Eisenbarth, T., (2020) Differential Power Analysis of the Picnic Signature Scheme. *Cryptology ePrint Archive Preprint.*
URL: https://eprint.iacr.org/2020/267.pdf

[13] *GitHub – Brazil Quantum*
URL: https://github.com/brazilquantum/PQC-Pix

[14] *Status Report on the Second Round of the NIST PQC Standardization Process*
URL: https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf

[15] *Microsoft Azure – Cloud Services*
URL: https://azure.microsoft.com/pt-br/

[16] *IonQ Takes Quantum Computing Public With A $2 Billion Deal*
URL: https://www.forbes.com/sites/moorinsights/2021/03/23/ionq-takes-quantum-computing-public-with-a-2-billion-deal/?sh=b62db775d062

[17] *Quantum Computing in the NISQ era and beyond*
URL: https://arxiv.org/abs/1801.00862

[18] *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*
URL: https://arxiv.org/abs/quant-ph/9508027

[19] *Post-Quantum Cryptography*
URL: https://www.springer.com/gp/book/9783540887010

[20] *Post-Quantum Cryptography: Current state and quantum mitigation*
URL: https://www.enisa.europa.eu/publications/post-quantum-cryptography-current-state-and-quantum-mitigation

[21] *Hash-based Signatures: An Outline for a New Standard*
URL: https://csrc.nist.gov/csrc/media/events/workshop-on-cybersecurity-in-a-post-quantum-world/documents/papers/session5-hulsing-paper.pdf

[22] *IBM's Roadmap for Scaling Quantum Technology*
URL: https://www.ibm.com/blogs/research/2020/09/ibm-quantum-roadmap/

[23] *Hardware Security Module (HSM) - DINAMO Networks*
URL: https://www.dinamonetworks.com/hardware-security-module-hsm/

[24] *Using Thales Luna HSMs with quantum-safe security to protect IoT*
URL: https://www.isara.com/downloads/solution_brief/ISARA_Quantum_Safe_Thales.pdf

[25] *Post-quantum crypto agility – Utimaco HSMs*
URL: https://hsm.utimaco.com/solutions/applications/post-quantum-crypto-agility/

[26] Kales, D., Zaverucha, G., (2020) Improving the Performance of the Picnic Signature Scheme. *Cryptology ePrint Archive Preprint.*
URL: https://eprint.iacr.org/2020/427.pdf